Deep Learning for Sequences – Assignment 2 – Part 4

Daniel Juravski – 206082323
Eyal Orbach 015369317

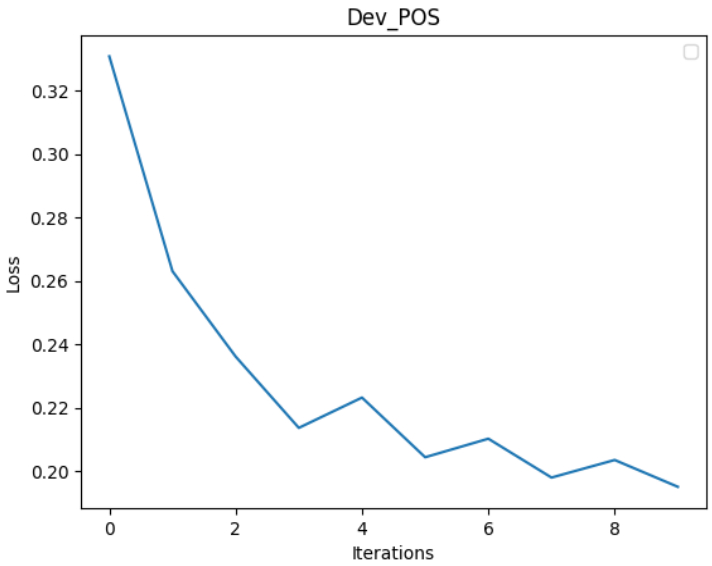
Our best parameters for both POS and NER:
Embedding vector size = 50
Ngram = 5
Hidden layer size = 50
Optimizer = “Adam”
Learning Rate = 0.002
Epochs = 5

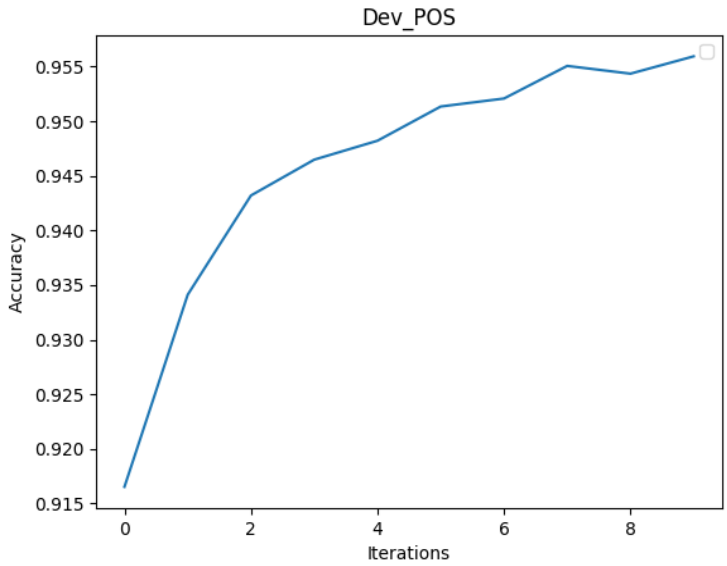
Considerations:

- We looped over our existing vocabulary of words (that we got from the train data or from the trained embedding matrix), we looked over every word there, checked if its' length bigger than 2 (deciding smaller words do not have significant prefix/suffix), and then parse the word to its' prefix of 3 letters and suffix of 3 letters and add it to the existing vocabulary.
- We had to make sure that those prefixes and suffixes getting a special signature and not overwriting existing words. For example the prefix of the word 'themselves' is 'the', we need to take care about the cases when we use the embedding of the prefix 'the' and when we use the embedding of the word 'the'.
- Like in part1, we had to care about the prefixes and suffixes that we getting on the train/dev data, but they do not appear in the embedding matrix. We create the ‘PRE_UUUNKKK’ and ‘SUFF_UUUNKKK’ tokens for representing it (those representations cannot be the same previous ‘UUUNKKK’ token because obviously there is a difference between an unknown word and unknown prefix and unknown suffix).
- To improve that, we have tried to add also prefix and suffix of 2 letters and 1 letter also. That not really worked for us, didn’t show some significant improvements.
- We found that the subword approach is more useful for that task, beacause of with that approach we can learn more behaviours about the words. On predict time, we look not only on the word (not surely that we got it in our embbeding matrix), but also on the preffix/suffix (more surely that we got it in our embbeding matrix), and can get clearer picture of the word features. The pre-trained vectors help, but we can get those pre-trained vectors values with some iterations over our data, so the sub-words defently were more useful.
- We found out that the subwords method worked best either for the POS data and the NER data as well. We found it reasonable because for example, for the POS data, the method has learned several prefixes and suffixes that can be dominant for POS tagging (suffixes like: ‘ing’, ‘ed’, ‘ly’ etc.).
- We found out that the best model performance is combination of subwords and pre-trained vectors.

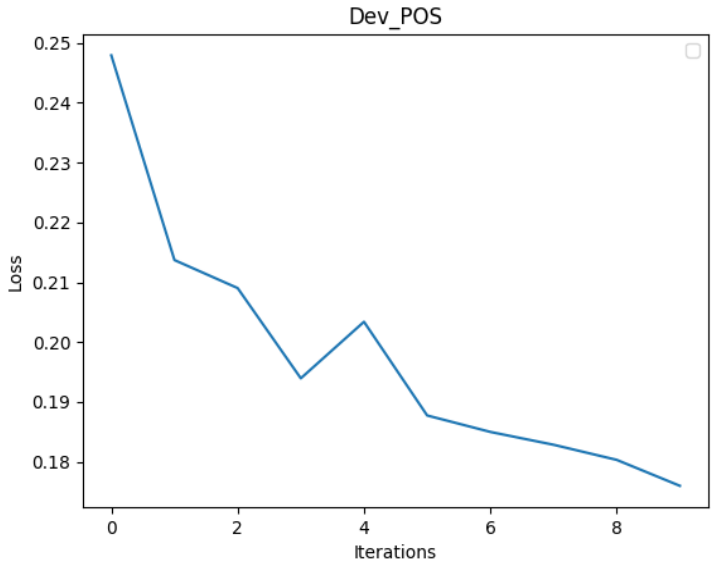
POS:

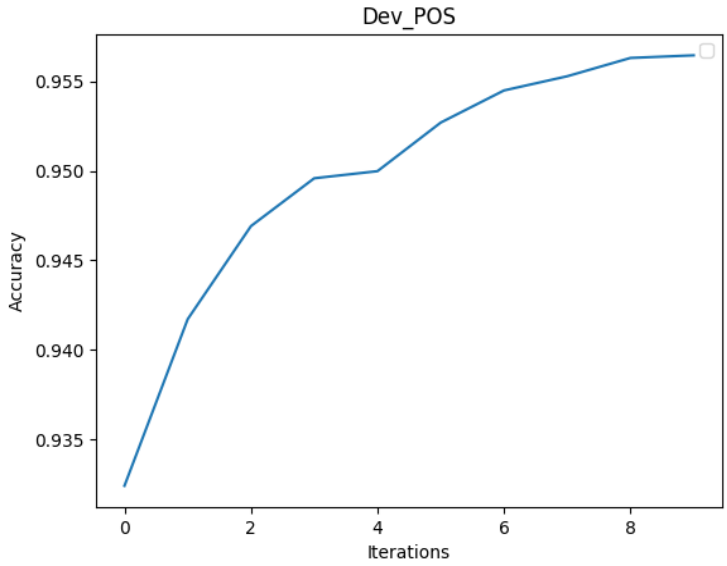
subword, no-embedding:





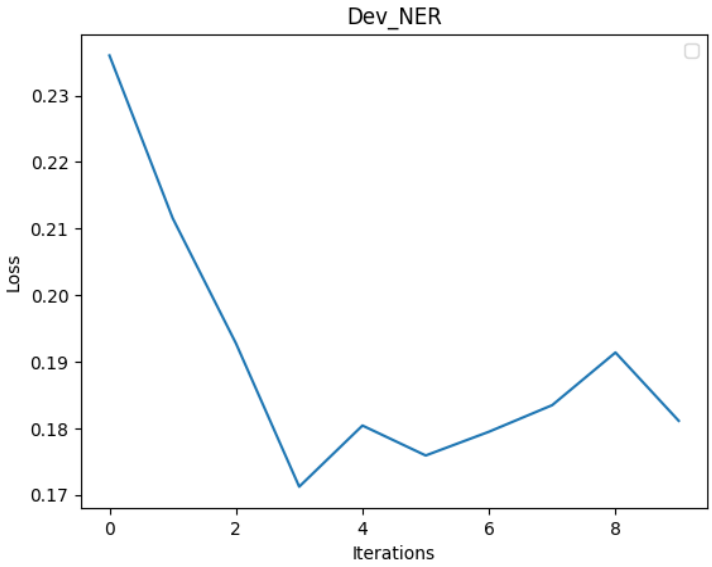
subword, embedding:

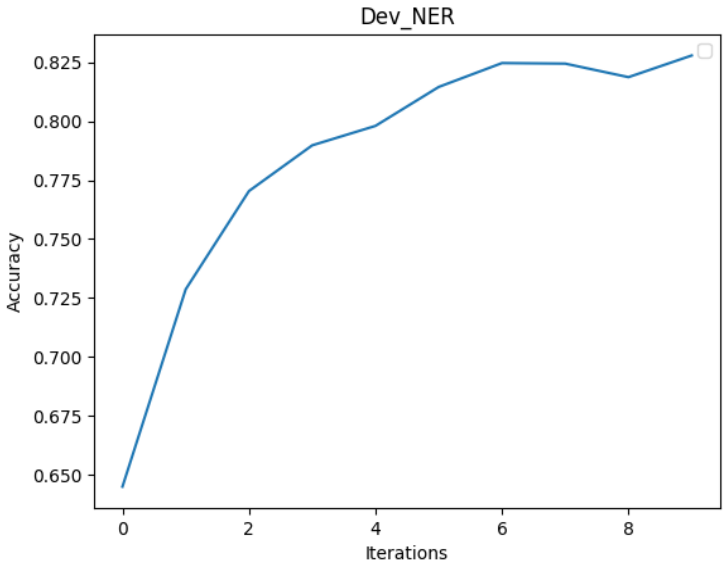




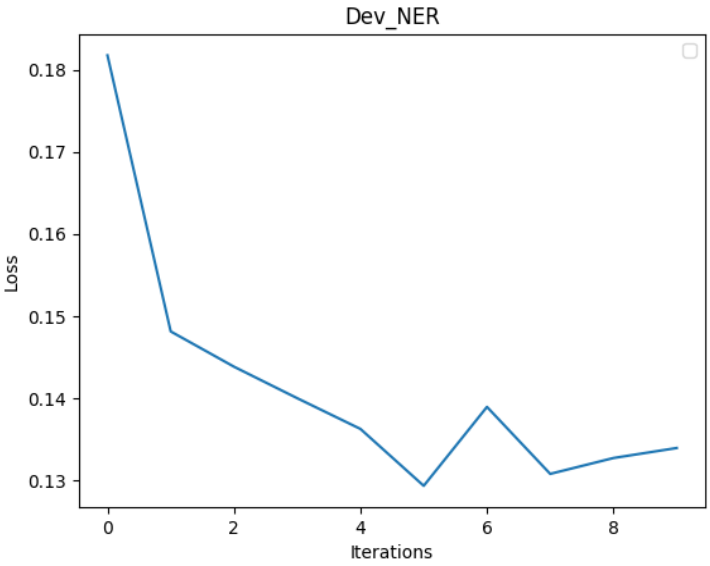
NER:

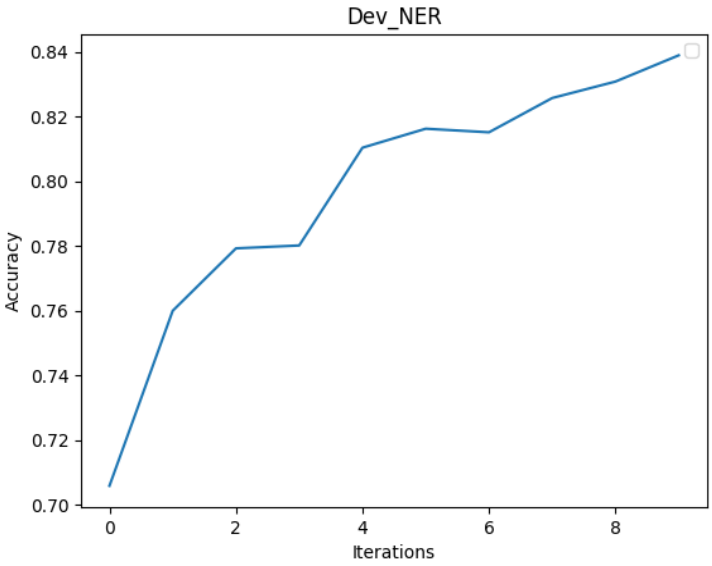
subword, no-embedding:





subword, embedding:





Notes:

1. Our model has learned in 5 epochs. We wanted to see the dev evaluation loss and accuracy some more times so we evaluate it twice in an epoch. Once in the middle of the epoch, and once at the end of the epoch.
2. The NER accuracy graph, calculated with the metric of ignoring the ‘O’ tag.